Claims

[c1] A method in a computer system for executing a program, the method comprising:

    providing a plurality of definitions of interactions of the program, an interaction having one or more command definitions defining commands and a view definition, a command having an attribute and a behavior, each interaction being associated with a request;

    receiving a request;

    identifying the interaction associated with the received request;

    for each command of the identified interaction,

        preparing the command by setting values of input attributes of the command based on attribute values stored in an attribute store;

        performing the behavior of the command with the input attributes; and

        extracting attribute values of the command by retrieving values of the output attributes of the command and storing the retrieved values of the attributes in the attribute store; and

    providing a response generated by the view associated with the identified interaction.

[c2] The method of claim 1 including before performing the behavior of the commands, setting values of attributes defined in the received request.

[c3] The method of claim 1 including before performing the behavior of a command, validating the values of the input attributes of the command.

[c4] The method of claim 1 wherein a command definition defines a value of an attribute.

[c5] The method of claim 1 wherein a command definition defines a scope associated with an attribute.

[c6] The method of claim 1 wherein the command definition defines an object

class and including instantiating an object of the object class before setting the values of the input attributes of the command.

[c7]   The method of claim 6 wherein the object class defines a Java bean.

[c8]   The method of claim 1 wherein the view definition defines a view with attributes and a behavior and wherein the performing of the view includes:

setting values of input attributes of the view based on attribute values stored in an attribute store; and

performing the behavior of the view with the set attributes.

[c9]   The method of claim 8 wherein the behavior of the view provides a display page.

[c10]   The method of claim 1 wherein the view definition defines a view with a target and including invoking the target to generate a response.

[c11]   The method of claim 10 wherein the response is a display page.

[c12]   The method of claim 11 wherein the target is a Java server page.

[c13]   The method of claim 11 wherein the target is an active server page.

[c14]   The method of claim 1 wherein the definition of an interaction includes a conditional that specifies a view to execute based on the value of an attribute in the attribute store.

[c15]   The method of claim 1 wherein the definition of an interaction includes a conditional that specifies a command whose behavior is to be performed based on the value of an attribute in the attribute store.

[c16]   The method of claim 1 wherein a command definition specifies mapping between a name of an attribute stored in the attribute store and a name of an attribute of the command.

[c17]

The method of claim 1 wherein a command definition indicates whether the behavior of the command should be performed when an error is detected

while performing the behavior of the commands of the interaction.

[c18]   The method of claim 1 including providing a translator for preparing command and for extracting attributes from the commands.

[c19]   The method of claim 1 wherein a command definition identifies a translator.

[c20]   The method of claim 1 wherein a command definition provides a value for an attribute that is used in place of the value of that attribute stored in the attribute store.

[c21]   A method in a computer system for executing a program, the method comprising:

providing a program with a plurality of interactions, the interactions specifying one or more commands, a command having one or more attributes and a behavior;

identifying interactions of the provided program to be performed; and

for each command specified by the identified interactions,

identifying an input attribute of the command;

setting a value of the identified input attribute based on the value of an output attribute of a previously performed command of an interaction of the provided program; and

performing the command to generate values of output attributes of the command in accordance with the behavior of the command.

[c22]   The method of claim 21 wherein the identifying of interactions includes receiving a request from a client indicating an interaction of the provided program that is to be performed.

[c23]   The method of claim 21 wherein the identifying of the input attribute of a command includes identifying a set method of an object implementing the command.

[c24]

The method of claim 21 wherein the identifying of the input attribute of a

command includes identifying a data member of an object implementing the command.

[c25]   The method of claim 21 including after performing a command storing a value of an output attribute of the command in an attribute store and wherein the setting of a value of the identified input attribute includes retrieving the value from the attribute store.

[c26]   The method of claim 21 including after performing a command, setting the value of each input attribute of each command of the interaction that corresponds to an output attribute of the performed command to the value of that output attribute.

[c27]   The method of claim 26 wherein each command includes an object that is instantiated before the value of an input attribute of the command is set.

[c28]   A method in a computer system for generating source code for a program, the method comprising:

    receiving a list of names of functions to be invoked by the program;
    for each of the functions to be invoked of the program,

        identifying names of formal parameters of the function; and
        outputting an invocation of the function that includes names of actual parameters derived from the identified names of formal parameters
        whereby the output invocations of the functions form the source code for the program.

[c29]   The method of claim 28 wherein each function has a signature that includes the name of the function and type of each formal parameter.

[c30]   The method of claim 28 wherein each function has an associated object and including outputting source code to instantiate the object for each function.

[c31]   The method of claim 30 wherein the object has a set and get method for setting and getting values of parameters.

[c32]   The method of claim 28 including receiving an alias for a name of a
parameter.

[c33]   A computer-readable medium containing a data structure defining a
command-based program, the data structure including:
    for commands of the program, a descriptor for the command, the
    descriptor identifying an object class, the object class defining input
    and output attributes and having a perform method
    whereby the program is executed by instantiating objects
    corresponding to the object classes of the commands, setting input
    attribute values of objects based on output attribute values of objects,
    and performing the perform method of the objects.

[c34]   The computer-readable medium of claim 33 wherein the commands are
organized into interactions.

[c35]   The computer-readable medium of claim 34 wherein each interaction has a
name.

[c36]   The computer-readable medium of claim 34 wherein the program operates
in request-response environment and a request specifies an interaction
whose commands are to be performed.

[c37]   The computer-readable medium of claim 36 wherein each interaction has an
associated view that specifies a response to be provided.

[c38]   The computer-readable medium of claim 37 wherein an interaction has a
condition for selection of a view.

[c39]   The computer-readable medium of claim 33 wherein the descriptor further
identifies an alias for an attribute of a command.

[c40]   The computer-readable medium of claim 33 wherein the descriptor further
identifies a constant for an attribute of a command.

[c41]
    The computer-readable medium of claim 33 wherein the descriptor further

identifies whether the command should be performed when an error is detected during execution of other commands of the program.

[c42]   A computer system for execution an interaction-based application, comprising:

means for receiving a request to perform an interaction of the application;

means for preparing each command of the requested interaction by identifying input attributes of the command and by setting input attribute values of the command;

means for performing the behavior of each command after the input attribute values of the command is set; and

means for retrieving output attribute values of commands after performing the behavior of a command so that the output attribute values are available for setting input attribute values of commands to be performed.

[c43]   The computer system of claim 42 including means for storing the retrieved output attribute values in an attribute store.

[c44]   The computer system of claim 42 including means for retrieving input attribute values from an attribute store.

[c45]   The computer system of claim 42 wherein the setting of an input attribute value occurs when a corresponding output attribute value is retrieved.

[c46]   The computer system of claim 42 wherein commands have objects with get and set methods and wherein the means for preparing a command includes after identifying the input attribute, invoking a get method of an object to retrieve an input attribute value.

[c47]   The computer system of claim 47 including after invoking the get method invoking a set method for the object of the command to set the input attribute value.

[c48]   A computer-readable medium containing instructions for controlling a computer system to execute a program, by a method comprising:

for each command of the program,

identifying input and output attributes of the command;

setting a value for each identified input attribute by retrieving from an attribute store;

performing a behavior of the command;

retrieving a value for an identified output attribute; and

storing the retrieved value in the attribute store.

[c49]   The computer-readable medium of claim 48 wherein the commands are organized into interactions.

[c50]   The computer-readable medium of claim 48 wherein each command has an object with methods for setting and getting values of attributes of the object.

[c51]   The computer-readable medium of claim 48 wherein each command has an object with a perform method for performing the behavior of the command.

[c52]   The computer-readable medium of claim 48 wherein a command includes an alias for an attribute.

[c53]   The computer-readable medium of claim 48 wherein a command include a value for an attribute.

[c54]   The computer-readable medium of claim 48 wherein a command has a translator for translating a value before setting an input attribute value.

[c55]   The computer-readable medium of claim 48 wherein the commands are organized into named interactions and the commands of an interactions are performed when a request specifying the name of the interaction is received.

[c56]   A method in a computer system for executing a program, the method comprising:

providing a plurality of interactions, an interaction having a command and another interaction having a view without having a command;

receiving an indication of an interaction of the program;

when the indicated interaction has a command,

setting input attributes of the command; and

performing a behavior associated with the command; and

when the indicated interaction has view, performing the view

whereby at least one interaction has a view without having any

commands.

[c57]   The method of claim 56 wherein when an interaction has two views, determining which view to perform.